LEAR – A Low-weight and Highly Adaptive Routing Method for Distributing Congestions in On-Chip Networks

Masoumeh Ebrahimi, Masoud Daneshtalab, Pasi Liljeberg, Juha Plosila, Hannu Tenhunen
Department of Information Technology, University of Turku
{masebr,masdan,pakrli,juplos,hanten}@utu.fi

Abstract-*Congestion-aware routing algorithms can improve network throughput by avoiding packets to be routed through congested areas. In this paper, we propose a minimal/non-minimal routing algorithm to alleviate congestion in the network by making use of all available paths between sources and destinations. The simplicity of the proposed algorithm provides a cost and power efficient solution for Networks-on-Chip while the high degree of adaptiveness, achieved by using an additional virtual channel along the Y dimension, leads to an increased performance. In this method, different restrictions are imposed on the use of each virtual channel, so that the prohibited turns in one virtual channel are permitted in the other one. By fully exploiting of the eligible turns in the network, a large number of output channels can be provided by the proposed method. Based on this method, a packet is routed along the non-minimal path when the neighboring routers in the minimal path are congested.*

1. Introduction

Network-on-Chip (NoC) has been widely researched and discussed as a candidate communication structure for complex MPSoCs due to its reusability and scalability [1][2]. The performance and efficiency of NoCs largely depend on the underlying routing technique which establishes a link between input and output channels in a router. In the routing process, the input selection and output selection are two key components of the router architecture. The input selection selects one of the input channels among all requesters to get access to an output channel. The output selection chooses an output channel among all potential output channels to deliver a packet. Routing algorithms could be classified as deterministic or adaptive [3]. A deterministic routing algorithm uses a fixed path for each pair of source and destination nodes. Adaptive routing algorithms have been proposed to meet performance specifications and to tolerate link or router failures. In adaptive routing algorithms, the path a packet travels from the source to the destination node is determined by network conditions. Hence, they can decrease the probability of routing packets via congested or faulty regions.

Adaptive routing algorithms can be either minimal or non-minimal. In minimal routing algorithms, shortest paths can be used for transmitting packets between source and destination. In non-minimal routing algorithms, packets can take longer paths and temporarily move away from the destination. Minimal adaptive routing algorithms that do not allow all packets to use any shortest path are called partially adaptive, while in fully adaptive methods, packets are able to choose among all the minimal paths available between the source and destination [5][6][7].

In this paper, we present an efficient minimal/non-minimal routing algorithm named Low-weight Extremely Adaptive Routing (LEAR) method to allow packets to be routed around congested areas. The proposed method is free from deadlock and livelock and it can provide a high degree of adaptiveness by utilizing one and two virtual channels along the X and Y dimensions, respectively. Unlike the existing non-minimal methods, the hardware overhead is low.

2. Related Work

Several minimal routing algorithms have been proposed in order to improve the performance of on-chip networks such as XY [6], West-First [7], Odd–Even [8], TM-FAR [9], DyAD [10], and BARP [11]. In [12] the locality decision is extended to two-hop neighbors. A well-known method named Regional Congestion Awareness (RCA) is proposed in [13] to utilize global congestion information in routing decision.

Virtual channels can be used both to avoid deadlock and increase adaptiveness. Three different minimal and fully adaptive routing algorithms based on a small number of virtual channels had been presented in [14], [15], and [17]. A desirable characteristic of these methods is using only two virtual channels along one of the two physical channels. However, since packets are limited to minimal paths, the traffic load cannot be balanced across the network. It has been proven that the mad-y algorithm [17] provides the maximal adaptiveness among the methods with the same number of virtual channels.

Several minimal and non-minimal approaches have been previously investigated in literature aiming to achieve fault-tolerant methods [18]-[22]. They are mostly proposed for supporting special cases of faults such as one-faulty router, convex or concave regions. Most of these techniques are based on adding a large number of virtual channels to avoid deadlock and increase the degree of adaptiveness. However, adding virtual channels can be expensive in terms of adding buffers and complex control logic to the routers. In general, each method defines a new tradeoff between the number of virtual channels and the ability to handle different fault models and increase adaptiveness, so that the larger number of virtual channels is, the more ability to support fault cases. The goal of our approach is to present a low restrictive minimal/non-minimal method by using only two virtual channels along one of the two dimensions. This method does not have the restrictions of fault-tolerant methods and the provided maximal adaptiveness by the algorithm permits packets to be routed through less congested areas, and thus alleviating the traffic load.

3. Preliminaries

The simplest deterministic routing algorithm routes packet by crossing dimensions in strictly decreasing order. This routing algorithm is very popular and is known as XY or YX algorithm in 2D mesh networks [16]. Fig. 1(a) shows the turns that are permitted to be taken in the XY routing algorithm. As can be seen in this figure, out of eight possible turns, four turns are prohibited while the other four turns are permitted [17]. Turn model is chosen as a representative of minimal and partial adaptive routing [17]. In the turn model, deadlock can be avoided by prohibiting enough turns to break all the cycles [3]. Three famous turn models are Negative-First (Fig. 1(b)), West-First (Fig. 1(c)) and North-Last (Fig. 1(d)). While the XY routing algorithm prohibits four turns to avoid deadlock, the turn models avoid only two turns.

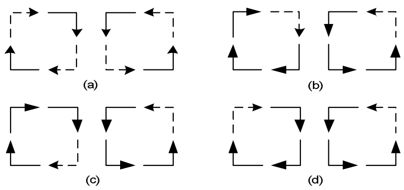

Fig. 1. All possible turns in (a) XY routing (b) Negative-First (c) West-First (d) North-Last (The solid lines indicate the permitted turns and the dash lines indicate the prohibited turns).

4. Minimal and Non-minimal Algorithms in Double-Y Network Using Turn Models

We propose a minimal/non-minimal routing algorithm, named Low-weight Extremely Adaptive Routing (LEAR) method, to alleviate congestion in the network. LEAR is based on the mad-y method which has been introduced by Glass and Ni in [17]. Both LEAR and mad-y algorithms utilize only two virtual channels along the Y dimension in 2D mesh network.

4.1 The mad-y Method

In 2D mesh network, three kinds of turns can be taken: 0-degree, 90-degree and 180-degree turns. 0-degree turns can occur if a packet travels in a same direction with a possibility of switching between virtual channels. Two types of 0-degree turns can be taken in a double-Y network (contains one virtual channel along the X dimension and two virtual channels along the Y dimension), 1- The turns that do not change the direction and the virtual channel of the packet. 2- The turns that do not change the packets direction but virtual channels. By taking a 90-degree turn, a packet can be transmitted between the nodes in perpendicular dimensions. Using a 180-degree turn, a packet arriving from one neighboring router will go back to that router [23].

In the mad-y routing algorithm, the first type of 0-degree turns is allowable as shown in Fig. 2 (c)). For the second type, since a packet switches between virtual channels, there is a possibility of deadlock. As it was proven in the mad-y method [17], the 0-degree turns from vc2 to vc1 may cause deadlock in the network, however the turns from vc1 to vc2 can be used without creating cycles (Fig. 2 (d)). As illustrated in Fig. 2 (a) and Fig. 2 (b), out of sixteen 90-degree turns that can be potentially taken in a network, four of them cannot be taken in the mad-y routing algorithm. 180-degree turns are not allowed in the mad-y method as they introduce a non-minimal routing. The mad-y method uses the same number of virtual channels as the two previously proposed fully adaptive algorithms in [17] and [18]. However, the ability of switching between virtual channels in the mad-y method provides an opportunity for choosing between three channels at a router (i.e. two virtual channels along the Y dimension and one along the X dimension), thus increasing the degree of adaptiveness as compared to [17] and [18].

When a packet enters a router through one of the input channels, the routing unit determines one or several potential output channels to deliver the packet. The routing decision is based on the relative position of the current node and the destination node that is within one of the following eight cases: north, south, east, west, northeast, northwest, southeast and southwest. Table 1 presents the choices of output channels allowed by the LEAR method. The table contents are listed based on a destination position (Position) and an input channel of the arriving packet (InCh).

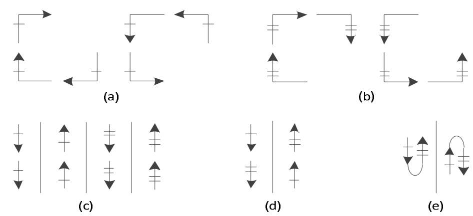

Fig. 2. All eligible turns in the mad-y (except (e)) and LEAR method.

4.2 The LEAR Method

In this section, we introduce a non-minimal and adaptive routing algorithm, named LEAR, which is based on the mad-y method. As the mad-y routing algorithm is a minimal routing method, it cannot fully utilize the eligible turns to route a packet through less-congested areas. The aim of the LEAR algorithm is to enhance the capability of the existing virtual channels for misrouting packets around congested areas and hotspots. In the mad-y and LEAR methods, the prohibited turns in two virtual channels are differentiated from each other. Therefore, they diminish the drawbacks of turn models that prohibit certain turns at all location. This is an important property of the LEAR method to place less restriction on routing packets in different directions.

Due to the minimal nature of the mad-y routing algorithm, all 180-degree turns are prohibited in this algorithm. However, they can be incorporated in non-minimal algorithms such as the proposed LEAR method. By a 180-degree turn or U turn, the direction of travel is reversed. We have examined 180-degree turns to check the possibility of deadlocks. 180-degree turns from vc1 to vc2 do not create cycles. However 180-degree turns from vc2 to vc1 defines a dependency among channels (Fig. 2(e)). Therefore, it is safe to employ 180-degree turns only from vc1 to vc2. All permitted 0-degree, 90-degree and 180-degree turns in the LEAR method are shown in Fig. 2.

Table 1. Possible output channels in the mad-y method based on the input channel (InCh) and relative position of source and destination (Position).

Position \ InCh	local	North-vc1	North-vc2	South-vc1	South-vc2	east	west
north	N1, N2	-	-	N1, N2	N2	N1, N2	N2
south	S2	S1, S2	S2	-	-	S2	S2
east	E	E	E	E	E	-	E
west	W	W	-	W	-	W	-
northeast	N1, N2, E	-	-	N1, N2, E	N2, E	N1, N2	N2, E
northwest	N1, W	-	-	N1, W	-	N1, W	-
southeast	S1, S2, E	S1, S2, E	S2, E	-	-	S1, S2	S2, E
southwest	S1, W	S1, W	-	-	-	S1, W	-

A node receiving a packet needs to check the eligible turns prior to connecting the input channel to the output channel. All permitted turns shown in Fig. 2 are not necessarily suitable choices for a packet as the packet may not be able to continue the path to the destination. It may occur for example when the destination of a packet is in the north and the packet makes an eligible North_vc1 to East turn or North_vc2 to East turn. In order to reach the destination, several turns must be taken by the packet in the remaining path that includes East to North_vc2 turn, North_vc2 to North_vc1 turn, and North_vc1 to West turn. Among them, the use of North_vc2 to North_vc1 turn is prohibited by the LEAR method. In this case, the algorithm never permits the packet to reach the destination, and thus the packet is blocked forever. In the LEAR method, the output channels are selected in a way that not only the turn connecting the input channel to the output channel is permissible but also it is guaranteed that there is at least one possible path from the next router to the destination. Table 2 shows the choices of output channels allowed by the LEAR method. One of the aims of the LEAR approach is to fully utilize all eligible turns to present a low restrictive adaptive method in the double-Y network. To achieve the maximal adaptiveness, for each combination of the input channel and destination node, we examined all 0-degree, 90-degree and 180-degree turns to find out the potential output channels. As can be obtained from Table 2, the LEAR method can offer a large number of output channels to route packets.

Since several output channels can be provided by the LEAR method, the output selection mechanism is needed to select an output channel among them. According to the LEAR method, if output channels in the minimal paths are congested, the routing unit uses the non-minimal path instead. To do this, at first, the output channels in the minimal path are examined and the packet is sent through the first output channel in which the corresponding downstream router has not raised its congestion flag. If the congestion flags of all neighboring routers in the minimal path are asserted, the congestion condition of the non-minimal paths is checked. If there is a non-minimal direction that is not congested, it is chosen as an output channel to deliver the packet. Fig. 3 shows an example of the LEAR method in 6×6 mesh network in which the source node at (1,1) sends a packet to the destination node at (4,3). According to Table 2, the packet arriving from the local channel and delivering toward the destination in northeast position has six alternative choices (i.e. N1,N2,S1,S2,E, and W); among them, the output channels N1, N2 and E introduce the minimal paths and S1, S2 and W indicate the non-minimal paths. Since the neighboring routers in the minimal paths are in the congested area, the packet is sent to the non-minimal direction that is not congested. If the west direction is selected, the packet at the next router will have five choices (i.e. N1,N2,S1,S2, and W). However, since the node (0,1) is located at a edge of the network, the west direction cannot be selected to deliver the packet. The node (0,2) in the north direction is in the minimal path and it is not congested, so it is chosen as the next hop. The same strategy is used until the packet reaches the destination node. This example shows the ability of the LEAR method to misroute packets around congested areas and balance the traffic load among alternative paths.

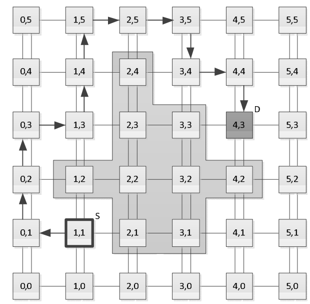

Fig. 3. An example of misrouting a packet around congested area using the LEAR method.

The non-minimal methods must be proved to be deadlock free and livelock free.

Theorem 1: the LEAR routing algorithm is deadlock free.

Proof: deadlock occurs when network resources continuously wait for each other to be released. If numbering mechanism assures that all eligible turns are ordered in ascending order (descending order), all packets has to travel along channels of strictly increasing (descending) numbers, so that no cyclic dependency can occur between channels and Theorem1 is proved. Like the mad-y algorithm, in the LEAR algorithm, a two-digit number (a, b) is assigned to each output channel. According to the numbering mechanism, a turn connecting the input channel (a_{ic}, b_{ic}) to the output channel (a_{oc}, b_{oc}) is called an ascending turn when ($a_{oc}>a_{ic}$) or (($a_{oc}=a_{ic}$) and ($b_{oc}>b_{ic}$)). As can be obtained from Fig. 4, all connections between input channels and output channels to form eligible turns in the LEAR method take place in ascending order. This ordering mechanism eliminates cycles in the channel dependency graph, and thus the LEAR method is proven to be deadlock free.

Theorem 2: The LEAR routing algorithm is livelock free

Proof: Livelock is a situation when packets circulating the network without any progress toward their destinations. In the LEAR method, whenever a packet transmits to the east direction, it never can be routed back to the west direction. Therefore, in the worst case, the packet may reach to the leftmost column and then start moving to the east direction toward the destination column. In this case, the packet may take virtual channel 1 and 2 in each column but then it has to take the east channel and make a progress toward the destination. Therefore, after a limited number of hops, the packet reaches the destination, and Theorem 2 is proved.

Table 2. Potential output channels according to the input channel and relative position of source and destination.

InCh / Position	**local**	**north-vc1**	**north-vc2**	**south-vc1**	**south-vc2**	**east**	**west**
north	N1, N2, S1, W	N2, S1, W	-	N1, N2, W	N2	N1, N2, S1, W	N2
south	N1, S1, S2, W	S1, S2, W	S2	N1, S2, W	-	N1, S1, S2, W	S2
east	N1, N2, S1, S2, E, W	N2, S1, S2, E, W	S2, E	N1, N2, S2, E, W	N2, E	N1, N2, S1, S2, W	N2, S2, E
west	N1, S1, W	S1, W	-	N1, W	-	N1, S1, W	-
northeast	N1, N2, S1, S2, E, W	N2, S1, S2, E, W	S2, E	N1, N2, S2, E, W	N2, E	N1, N2, S1, S2, W	N2, S2, E
northwest	N1, S1, W	S1, W	-	N1, W	-	N1, S1, W	-
southeast	N1, N2, S1, S2, E, W	N2, S1, S2, E, W	S2, E	N1, N2, S2, E, W	N2, E	N1, N2, S1, S2, W	N2, S2, E
southwest	N1, S1, W	S1, W	-	N1, W	-	N1,S1, W	-

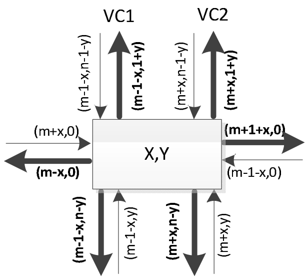

Fig. 4.LEAR method numbering mechanism

5. Results and Discussion

In this section, we assess performance of different routing algorithms using, a cycle-accurate NoC simulator developed in VHDL. As performance metrics, we use throughput and average delay. The simulator inputs include the array size, the router operation frequency, the routing algorithm, the link width length, and the traffic type. The simulator can generate different traffic profiles. The experiments are performed on a 2D-mesh 8×8 network using wormhole switching with a constant packet size of 8 flits. For all routers, the data width is set to 32 bits (the maximum bandwidth at each link is 1 flit per cycle) and each input channel has a buffer (FIFO) size of 12 flits with the congestion thresholds at 75% of the total buffer capacity. Two synthetic traffic profiles including uniform and hotspot are used to evaluate the LEAR method. As performance metrics, we choose throughput and delay. Throughput is measured as the fraction of the maximum load that the network is capable of physically handling. Latency defined as the number of cycles between the initiation of a packet transmission issued by a source node and the time when the packet is completely delivered to the destination node. The time needed to generate packets is not considered, because we assumed the packets are generated in the processing elements. For each load value, the result of packet latency and throughput are averaged over 80,000 packets after a warm-up session of 20,000 arrived packets.

5.1 Performance Evaluation

For each traffic profile and routing algorithm, the average communication latency and throughput with various packet injection rates are computed. The routing schemes compared to LEAR are XY [6], DyXY [14], and mad-y. XY represents a deterministic scheme while DyXY and mad-y designate as fully adaptive minimal schemes.

Uniform Traffic Profile: In uniform traffic, a node sends the packet to other nodes with the same probability. Fig. 5 and Fig. 6 shows the latency and throughput results obtained from the network under the uniform traffic profile. As can be seen from the results, the XY outperforms the minimal adaptive routing schemes because of two reasons. First, XY embodies global long-term information about the uniform traffic profile spreading uniform traffic as evenly as possible across the channels [7][12]. Since the minimal adaptive algorithms employ local short term information, they may use zigzag paths, which disturb the global long-term evenness of uniform traffic which decrease the performance at higher injection rates [7][8][12]. Second, packets use minimal paths so that under this traffic they are routed through the very center of the network which creates large permanent hotspots in the center of the network. Correspondingly, packets traveling through the center of the network will be delayed much more than they would use any non-minimal paths. Due to the fact that the LEAR method can misroute the congested areas, it performs as well as XY. Using minimal and non-minimal routes augments the throughput of the presented routing scheme.

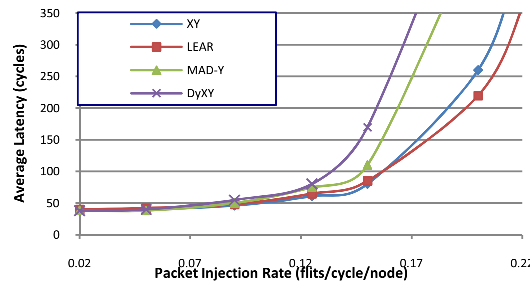

Fig. 5. Average latency under uniform traffic profile

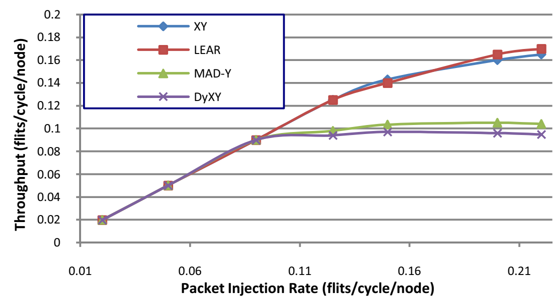

Fig. 6. Throughput under uniform traffic profile

Hotspot Traffic Profile: Hotspot is considered to be a more realistic traffic profile since processors often communicate with a part of the total number of other processors like memory modules. In the hotspot traffic pattern, one or more nodes are designated as hotspot nodes receiving an extra portion of the traffic in addition to the regular uniform traffic. Newly generated packets are directed to each hotspot node with an additional *H* percent probability. We simulate hotspot traffic with four hotspot nodes. Four hotspot nodes are chosen at the center of the mesh, (3, 3), (4, 3), (3, 4), (4, 4), with equal probability of *H*=20%. The average latency and throughput results are illustrated in Fig. 7 and Fig. 8, respectively. As the figures show, the proposed scheme achieves both higher throughput and lower latency. This is due to the good ability of routing around local congestion, which makes up for its difficulty in handling the central network-induced mesh hotspots.

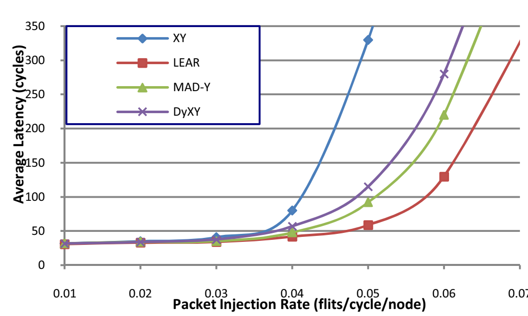

Fig. 7. Average latency under hotspot traffic profile

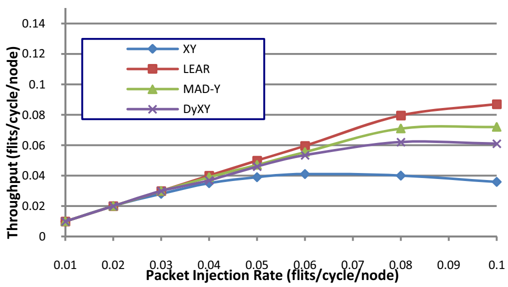

Fig. 8. Throughput under hotspot traffic profile

5.2 Physical Analysis

To analyze the physical implementation of each routing scheme four different on-chip networks, LEAR, mad-y, DyXY, and XY, are formed. The whole platform of each network including network interfaces, routers, and communication channels is synthesized by Synopsys D.C. using the UMC 90nm technology with an operating point of 1GHz and supply voltage of 1V. We performed place-and-route, using Cadence Encounter, to have precise power and area estimations. The power dissipation of each scheme is calculated under the hotspot traffic near the saturation point using Synopsys PrimePower in a 6×6 2D mesh. The layout area and power consumption of each platform are shown in Table 3.Comparing the area cost of the platform using the LEAR method with other platforms indicates that the LEAR platform imposes less than 1% hardware overhead in comparison with the minimal adaptive platforms, mad-y and DyXY. Since the XY platform does not require any virtual channel in north and south dimensions the area cost is 14% smaller than the LEAR platform. The power consumption of each platform under hotspot traffic near the saturation point (5%) is reported in Table 3. The LEAR platform consumes more average power because of misrouting packets around the congestion areas which increases the hop counts. To illustrate how the presented approach reduces the network hotspots, the maximum power value of each platform is also reported in the table. The results indicate that the maximum power of the presented approach is 8%, 11%, and 12% less than that of the XY, DyXY, and mad-y platforms, respectively. This is achieved by smoothly distributing the power consumption over the network using the adaptive routing scheme which reduces the number of the hotspots.

Table 3. Hardware implementation details.

Platforms	Area (mm^2)	Avg. Power (W)	Max. Power (W)
XY	5.722	2.21	3.32
DyXY	6.670	2.34	3.42
mad-y	6.701	2.41	3.46
LEAR	6.803	2.75	3.05

6. Conclusion

In this paper, we have proposed a minimal/non-minimal routing algorithm in 2D mesh Networks-on-Chip. The algorithm requires one and two virtual channels along the X and Y dimensions. To relax the restriction of the turn model, the prohibited turns on the first virtual channel is defined different from the second one. The presented method provides a high degree of adaptiveness to allow packets to be routed around congested areas. It can be easily shown that the LEAR algorithm can support all one-faulty cases and some multiple link/node failures. So, the LEAR method can efficiently avoid congestion in non-faulty networks, while it can be reconfigured to support faulty patterns with a lower degree of adaptiveness when faults occur in the network. This can be discussed and investigated in our future work.

7. Acknowledgment

The authors wish to acknowledge the Academy of Finland and Elisa Foundation for the partial financial support of this research.

References

[1] A. Jantsch and H. Tenhunen, “Networks on Chip,” Kluwer Academic Publishers, 2003.
[2] M. Daneshtalab, and et al., “A Low-Latency and Memory-Efficient On-hip Network,” in Proceedings of 4th International Symposium on Network-on-Chip (NOCS), pp. 99-106, May 2010, France.
[3] J. Duato, et al., “Interconnection networks: an engineering approach,” Morgan Kaufmann Publishers, 2003.
[4] L.M. Ni and et al., “A survey of wormhole routing techniques in direct networks,” Proc. IEEE Computer, v.26, I.2, pp.62-76, 1993.
[5] W. J. Dally and B. Towles, “Principles and Practices of Interconnection Networks,” Morgan Kaufmann, 2004.
[6] Intel Corporation, A touchstone delta system description, in: Intel Advanced Information, 1991.
[7] C.J. Glass et al., “The Turn Model for Adaptive Routing,” proc. 19th Ann. Int'l Symp. Computer Architecture, pp. 278-287, 1992.
[8] G. Chiu, “The Odd-Even Turn Model for Adaptive Routing,” IEEE Tran. On Parallel and Distributed System, pp. 729-738, 2000.
[9] T. Wen-Chung, et al., “TM-FAR: Turn-Model Based Fully Adaptive Routing for Networks on Chip,” Proc. VLSI-SoC, pp. 19-24, 2010.
[10] J. Hu and R. Marculescu, “DyAD-Smart Routing for Networkson-Chip,” Proc. DAC, pp. 260-263, 2004.
[11] P. Lotfi-Kamran, et al., “BARP- A Dynamic Routing Protocol for Balanced Distribution of Traffic in NoCs to Avoid Congestion,” in Proceedings of 11th IEEE/ACM Design, Automation, and Test in Europe (DATE), pp. 1408-1413, Mar 2008, Germany.
[12] G. Ascia, et al., “Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip,” IEEE Transaction on Computers, v.57, I.6, pp. 809-820, 2008.
[13] P. Gratz, B. Grot, et al., “Regional Congestion Awareness for Load Balance in Networks-on-Chip,” Proc. HPCA, pp. 203-214, 2008.
[14] M. Li, Q.A. Zeng, et al., “DyXY – a proximity congestion-aware deadlockfree dynamic routing method for network on chip,” Proc. DAC, pp. 849-852, 2006.
[15] M. Ebrahimi, M. Daneshtalab, et al, “Agent-based On-Chip Network Using Efficient Selection Method,” in Proceedings of 19th IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), pp. 284-289, Oct 2011, Hongkong.
[16] M. Dehyadegari, et al, “An Adaptive Fuzzy Logic-based Routing Algorithm for Networks-on-Chip,” in Proceedings of 13th IEEE/NASA-ESA International Conference on Adaptive Hardware and Systems (AHS), pp. 208-214, June 2011, USA.
[17] C. Glass and L. Ni, ”Maximally Fully Adaptive Routing in 2D Meshes,” Proc. Parallel Processing, pp.101-104, 1992.
[18] W.J. Dally and H. Aoki, “Adaptive Routing Using Virtual Channels,” Tech. Rep. MIT, Lab. of CS, Cambridge, Massachussetts, 1990.
[19] F. Chaix, et al., “A fault-tolerant deadlock-free adaptive routing for On Chip interconnects,” Proc. DATE, pp. 1-4, 2011.
[20] M. Valinataj, et al., “A reconfigurable and adaptive routing method for fault-tolerant mesh-based networks-on-chip,” AEU - International Journal of Electronics and Communications, v. 65, I. 7, pp.630-640, 2011.
[21] Z. Zhang, et al., “A reconfigurable routing algorithm for a fault-tolerant 2d-mesh network-on-chip,” in Proc. DAC, 2008.
[22] S. Jovanovic, C. Tanougast, et al., “A new deadlock-free fault-tolerant routing algorithm for NoC interconnections,” Proc. FPLA, pp.326-331, 2009.
[23] K.P. Fan, C.T. King, “Turn Grouping for Multicast in Wormhole-Routed Mesh Networks Supporting the Turn Model,” Journal of Supercomputing, v.16, No.3, pp. 237 - 260, 2000.